Network Service Security Through Software-Defined Networking

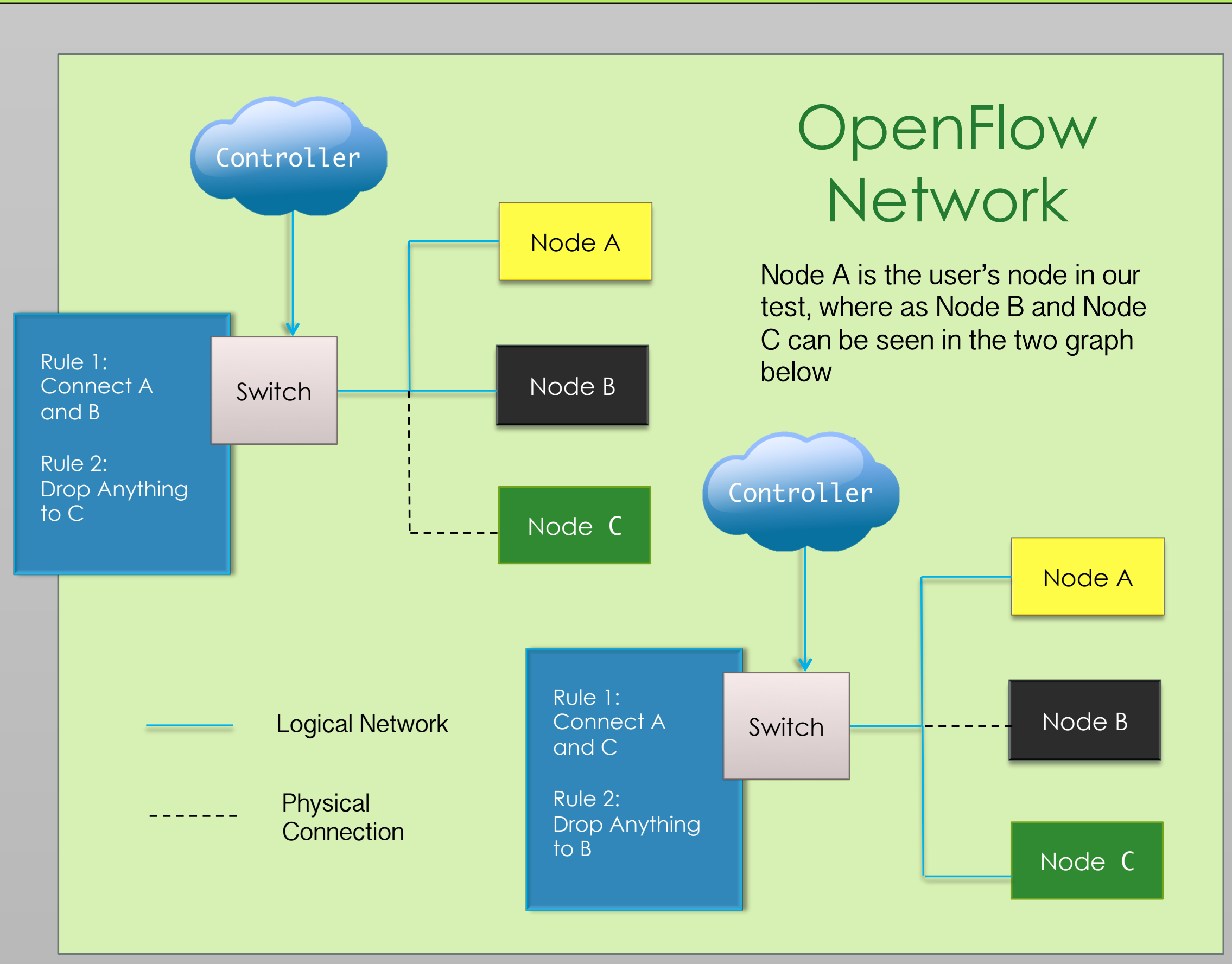
John Bilberry (University of Texas at El Paso), Melanie Palmer (New Mexico Tech), and Rob Sullivan (Michigan Tech)
Mentors: Kyle Lamb (HPC-3) and Ben McClelland (HPC-5)
Instructor: Dane Gardner, assisted by Matt Broomfield

Introduction

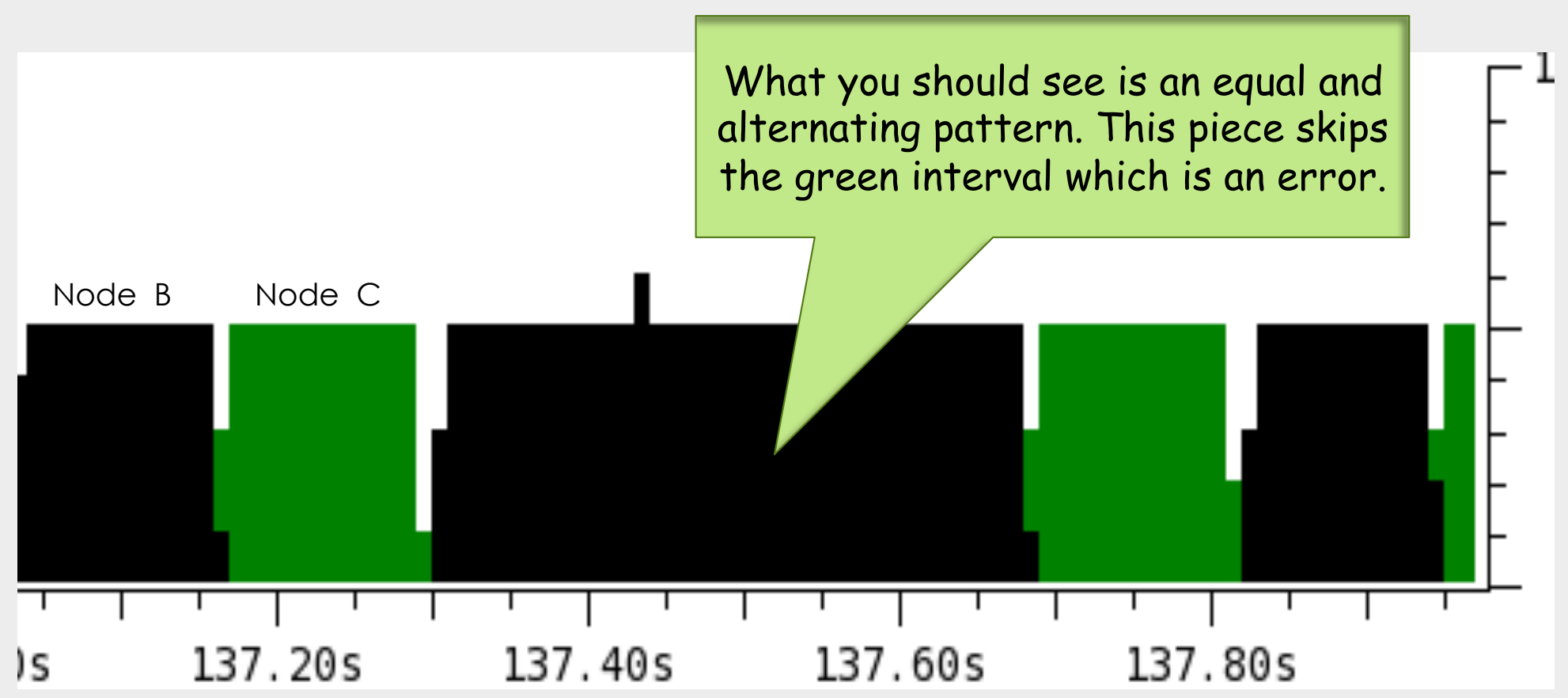
When working with job resource schedulers on large clusters, especially those with sensitive or classified data, security is a major concern. OpenFlow is an open standards communication protocol that gives network administrators the ability to configure networks through software in a way that can limit traffic to and from specific parts of a network. This offers a significant advantage over manual configuration of switches, a process that can be complicated and varies between switch manufacturers. LANL jobs are often spread out over many nodes, so the time it takes to add or modify network traffic rules is directly related to the security of the information on the rest of the cluster.

Objective

Our goal is to assess the performance, reliability and scalability of OpenFlow. We will examine how quickly and reliably network configurations can be changed in order to determine if OpenFlow could be utilized on the scale of the clusters at LANL.


OpenFlow Network

Node A is the user's node in our test, where as Node B and Node C can be seen in the two graph below


What you should see is an equal and alternating pattern. This piece skips the green interval which is an error.

Materials

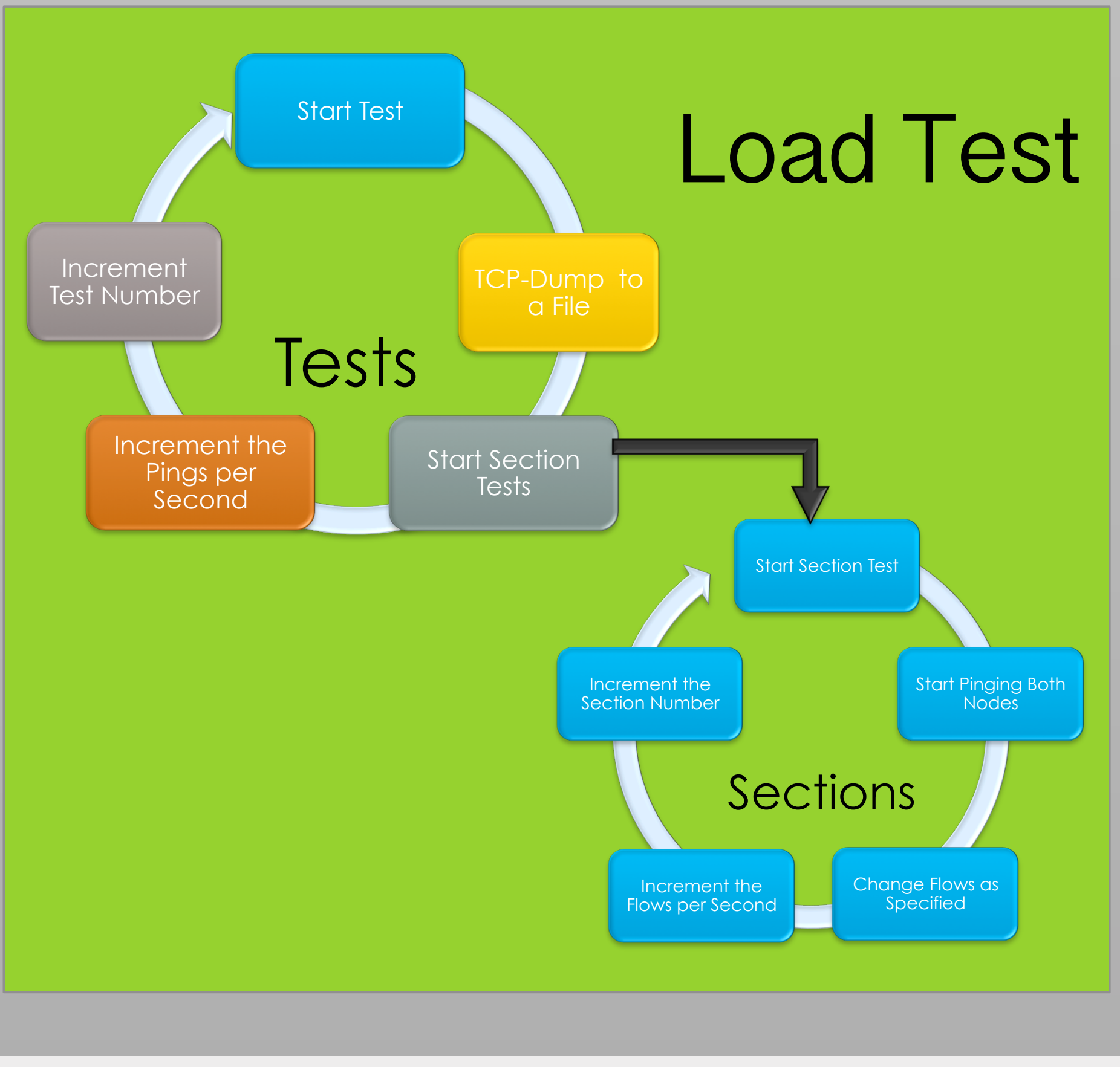
- Seven node CentOS 6.4 cluster
- Arista 7050S switch meeting OpenFlow 1.0 specification
- Floodlight 0.9 controller
- Custom testing suite

Terminology

- Software Defined Networking (SDN) –Adds a layer of management software between the switch and the administrator
- OpenFlow – Open source SDN protocol
- Controller – Software that communicates with the switch via a secure channel
- Flow – A rule given by the controller to the switch to direct network traffic
- Flow Error - When a flow update isn't pushed correctly or is missed

Test Suite

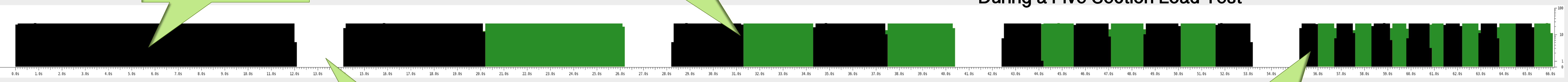
We designed the Load Test, described in the figure below, to find how fast flows could be pushed without error. The Speed Test finds the time required for a flow to be updated. This test sends traffic continuously to a single node while regularly pushing flows to the switch and records the time until the updates take effect.



This is our base case! This is what packets should look like when they are allowed to reach Node B.

This change in color is when the network is told to not allow traffic to Node B, but to let traffic through to Node C.

Network Traffic Displaying Flow Changes
During a Five Section Load Test

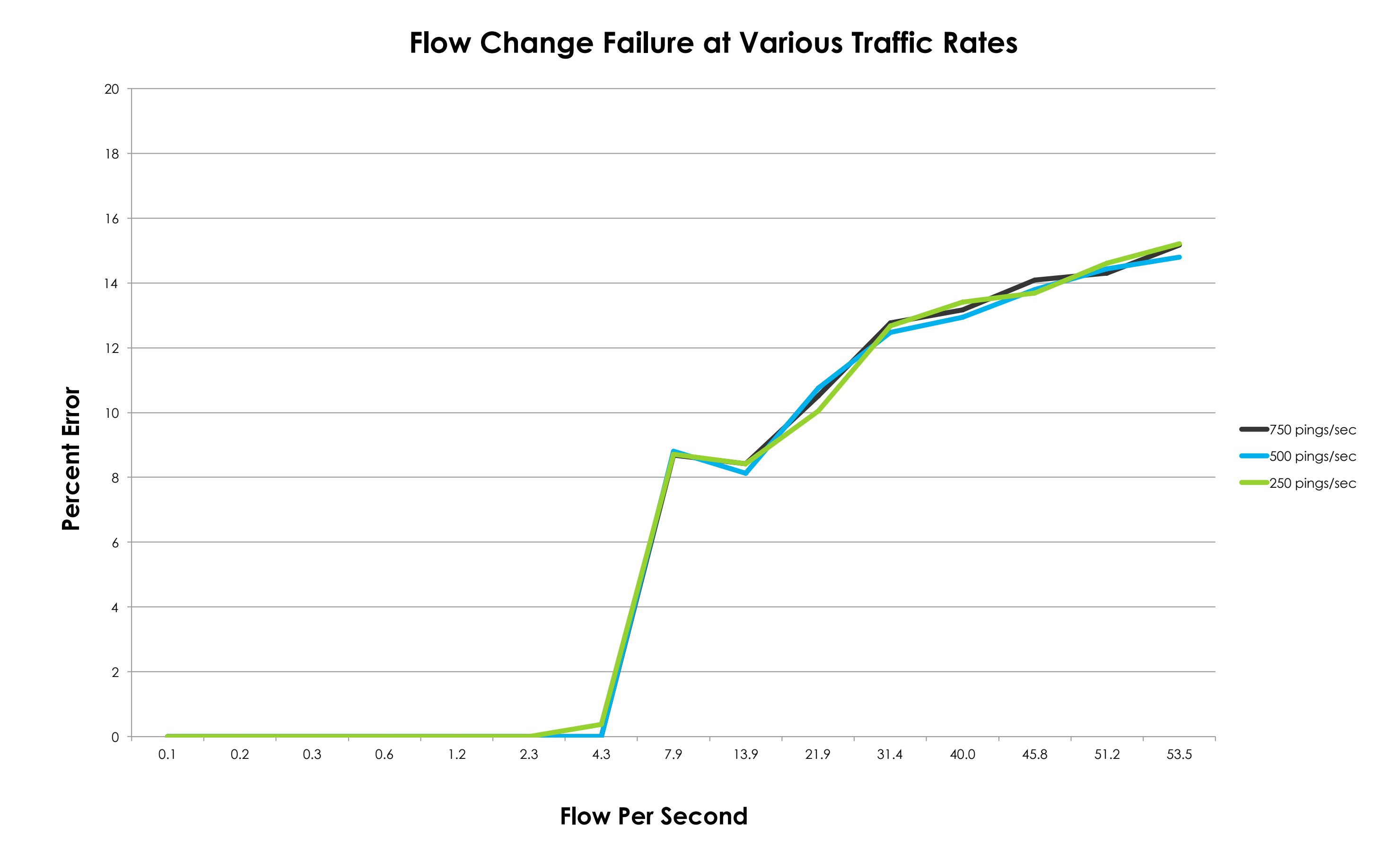


Don't fret! This white space is a rest between 'sections' so you can read it!

This is exactly what we were hoping to see!! They are switching perfectly!

Load Test

- Method
  - Three tests performed to determine flow failure rate, conducted at 250, 500, and 750 pings per second
  - Tests covered15 flow rates from 1 flow per 10 seconds to about 50 flows per second
  - Tests performed 10 times each to get reasonable statistics
- Expectations
  - To find a clear threshold at which flow pushes begin to fail
  - Switch traffic will affect the flow push speeds
- Results
  - System maintains complete reliability up to 5 flows per second
  - Reliability slowly diminishes at higher flow rate
  - Approximately 85% of flows push reliably at 50 flows per second
  - Switch traffic has no noticeable effect on flow push speed



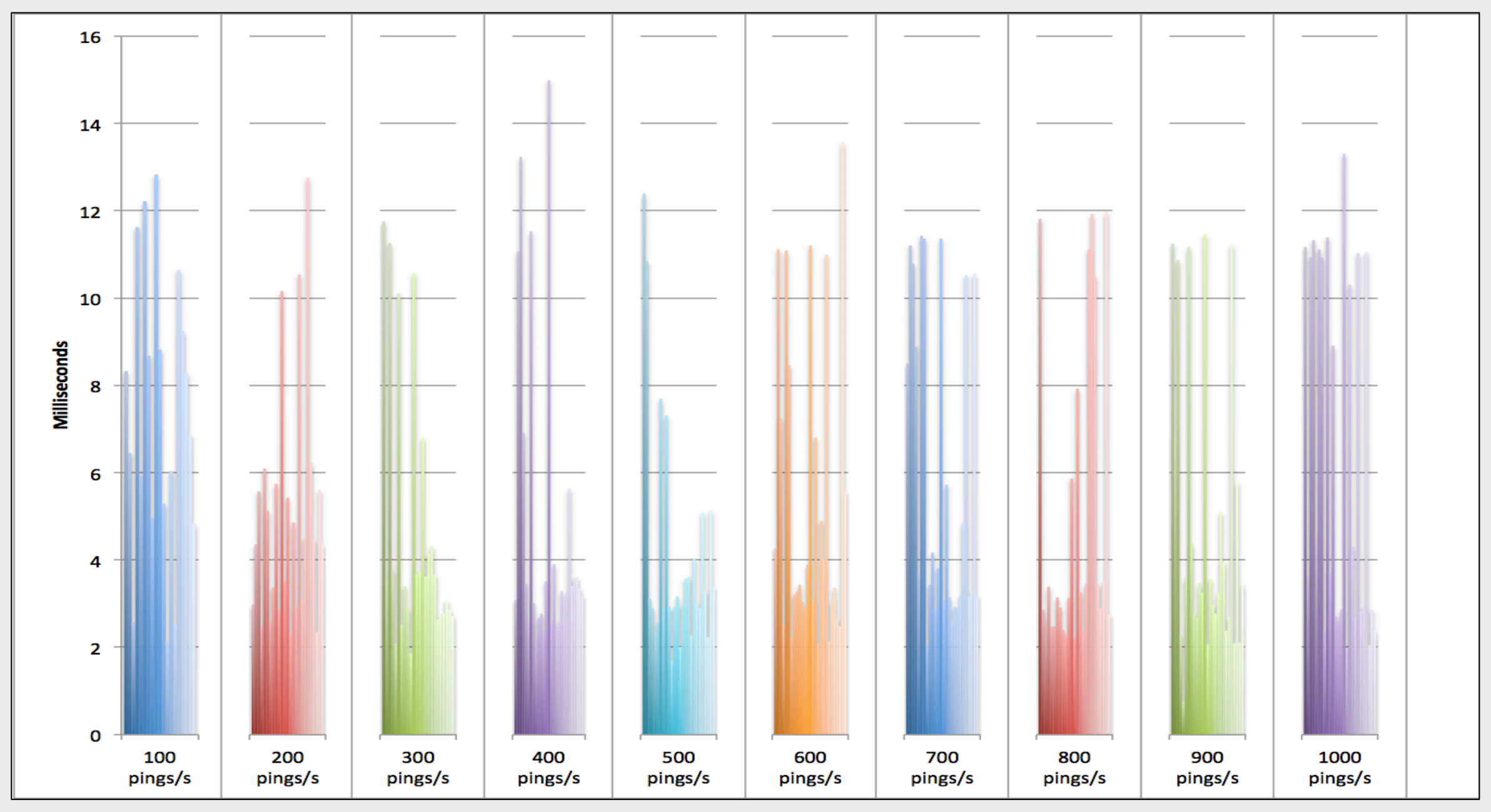
Conclusion

From performance testing on our specific hardware, we observed that OpenFlow allows networks to be dynamically reconfigured at up to 5 flows per second without any data loss. Configurations can be made at higher rates, but reliability suffers. We also discovered that it takes on average 5 ms from the time a flow is pushed to the time it is installed on the switch. The combination of the results from the Speed and Load tests suggest that the primary traffic bottleneck may be the controller. OpenFlow shows potential for use on the large clusters at LANL, but we recommend further scaled testing to see if flows for cluster jobs can be installed without significant increases in setup overhead.

Speed Test

- Method
  - Ten tests performed at rates from 100 to 1000 pings per second
  - Update flows once per second to alternately accept and drop traffic
  - Measure the time from when a flow is pushed to when the first packet matches on the new flow
- Results
  - Mean time to install a flow is 5.2 ms, standard deviation of 3.4 ms over 250 tests
  - Flow updates took as little as 0.76 ms and as long as 15 ms
  - Results reflect the test method; higher traffic equates to higher resolution for capturing faster flow updates
  - Time variations may be due to controller or switch buffering

Flow Update Speeds at Various Traffic Rates



Future Work

- Test expanded capabilities of latest version of the OpenFlow protocol
- Examine possible security issues with Floodlight controller
- Experiment with other OpenFlow controllers to find their strengths and weaknesses compared to Floodlight
- Test FlowVisor, a special purpose controller allowing the use of multiple OpenFlow controllers for a single network

Contact Us at:
Rob Sullivan: rssulliv@mtu.edu
Melanie Palmer: mpalmer@nmt.edu
John Bilberry: jsbilberry@ieee.org

LA-UR-13-25962